

CLAIMS

*Sub
ai*

1. A method for providing at least one self-tuning object to a user program, comprising:

- 5 receiving said user program;
 simulating execution of said user program;
 detecting, during said simulating of said execution of said user program, a plurality of expressions including said self-tuning object in said user program;
10 generating, in response to said detecting said plurality of expressions including said self-tuning object in said user program, a trace file indicating a sequence of said expressions including said self-tuning object in said user program;
15 dividing said trace file into a plurality of trace file blocks;
 converting said trace file blocks into source code expression blocks;
 generating a plurality of minimal timing, compiled
20 expression blocks, each of said plurality of minimal timing, compiled expression blocks corresponding to a respective one of said source code expression blocks, said generating said plurality of minimal timing, compiled expression blocks including application of at least one compiler optimization
25 technique; and
 linking said plurality of minimal timing, compiled expression blocks into said user program.

30 2. The method of claim 1, wherein said detecting said plurality of expressions including said self-tuning object in said user program is performed by program code associated

Sub
Doc
with at least one overloaded operator associated with said self-tuning object.

3. The method of claim 1, wherein said generating a trace
5 file indicating the sequence of said expressions including
said self-tuning object in said user program is performed by
program code associated with at least one overloaded
operator associated with said self-tuning object.

10 4. The method of claim 1, wherein said dividing said trace
file into said plurality of trace file blocks is performed
such that a total amount of computational dependencies and
synchronization requirements within said user program,
including computational dependencies and synchronization
15 requirements between trace file blocks, are minimized.

5. The method of claim 1, wherein said dividing said trace
file into said plurality of trace file blocks is performed
responsive to user provided delimiters included within said
20 user program.

6. The method of claim 1, wherein said generating said
plurality of minimal timing, compiled expression blocks
further comprises compiling and executing at least one of
25 said expression blocks multiple times while varying a value
of at least one optimization parameter for said at least one
compiler optimization technique.

7. The method of claim 6, wherein said generating said
30 plurality of minimal timing, compiled expression blocks

further comprises timing said multiple executions of said compiled expression blocks.

8. The method of claim 1, wherein said linking of said minimal timing, compiled expression blocks to said user program is responsive to execution of said user program.

9. The method of claim 8, wherein said linking of said minimal timing, compiled expression blocks further comprises detecting, during said execution of said user program, said plurality of expressions including said self-tuning object in said user program.

10. The method of claim 9, wherein said linking of said minimal timing, compiled expression blocks further comprises scheduling said minimal timing, compiled expression blocks for execution on at least one processor of a target parallel processing computer.

11. A computer program product including a computer readable medium, said computer readable medium having at least one computer program stored thereon, said at least one computer program comprising:

program code for receiving said user program;

program code for simulating execution of said user program;

program code for detecting, during said simulating of said execution of said user program, a plurality of expressions including said self-tuning object in said user program;

Sub
program code for generating, in response to said detecting said plurality of expressions including said self-tuning object in said user program, a trace file indicating a sequence of said expressions including said self-tuning
5 object in said user program;

program code for dividing said trace file into a plurality of trace file blocks;

program code for converting said trace file blocks into source code expression blocks;

10 program code for generating a plurality of minimal timing, compiled expression blocks, each of said plurality of minimal timing, compiled expression blocks corresponding to a respective one of said source code expression blocks, said generating said plurality of minimal timing, compiled
15 expression blocks including application of at least one compiler optimization technique; and

program code for linking said plurality of minimal timing, compiled expression blocks into said user program.

20 12. The computer program product of claim 11, wherein said program code for detecting said plurality of expressions including said self-tuning object in said user program comprises program code associated with at least one overloaded operator associated with said self-tuning object.

25 13. The computer program product of claim 11, wherein said program code for generating a trace file indicating the sequence of said expressions including said self-tuning object in said user program comprises program code
30 associated with at least one overloaded operator associated with said self-tuning object.

Sub
14. The computer program product of claim 11, wherein said program code for dividing said trace file into said plurality of trace file blocks is operative to divide said
5 trace file into said plurality of trace file blocks such that a total amount of computational dependencies and synchronization requirements within said user program, including computational dependencies and synchronization requirements between trace file blocks, are minimized.

10 15. The computer program product of claim 11, wherein said program code for dividing said trace file into said plurality of trace file blocks is operative to divide said trace file into said plurality of trace file blocks
15 responsive to user provided delimiters included within said user program.

20 16. The computer program product of claim 11, wherein said program code for generating said plurality of minimal timing, compiled expression blocks further comprises program code for compiling and executing at least one of said expression blocks multiple times while varying a value of at least one optimization parameter for said at least one compiler optimization technique.

25 17. The computer program product of claim 16, wherein said program code for generating said plurality of minimal timing, compiled expression blocks further comprises program code for timing said multiple executions of said compiled
30 expression blocks.

Sub
Par

18. The computer program product of claim 11, wherein said program code for linking of said minimal timing, compiled expression blocks to said user program is triggered by execution of said user program.

5

19. The computer program product of claim 18, wherein said linking of said minimal timing, compiled expression blocks further comprises program code for detecting, during said execution of said user program, said plurality of expressions including said self-tuning object in said user program.

10

20. The computer program product of claim 19, wherein said program code for linking of said minimal timing, compiled expression blocks further comprises program code for scheduling said minimal timing, compiled expression blocks for execution on at least one processor of a target parallel processing computer.

15

20 21. The computer program product of claim 11, wherein said computer program comprises a compiler.

22. A computer data signal embodied in a carrier wave, said computer data signal including at least one computer

25 program, said at least one computer program comprising:

program code for receiving said user program;

program code for simulating execution of said user program;

30 program code for detecting, during said simulating of said execution of said user program, a plurality of

Sub
expressions including said self-tuning object in said user program;

program code for generating, in response to said detecting said plurality of expressions including said self-tuning object in said user program, a trace file indicating a sequence of said expressions including said self-tuning object in said user program;

program code for dividing said trace file into a plurality of trace file blocks;

program code for converting said trace file blocks into source code expression blocks;

program code for generating a plurality of minimal timing, compiled expression blocks, each of said plurality of minimal timing, compiled expression blocks corresponding to a respective one of said source code expression blocks, said generating said plurality of minimal timing, compiled expression blocks including application of at least one compiler optimization technique; and

program code for linking said plurality of minimal timing, compiled expression blocks into said user program.

23. A system for providing at least one self-tuning object to a user program, comprising:

at least one processor;

at least one memory communicably coupled to said at least one processor;

a computer program for execution on said processor, said computer program stored in said memory, said computer program comprising:

program code for receiving said user program;

*Sub
Pat*

program code for simulating execution of said user program;

program code for detecting, during said simulating of said execution of said user program, a plurality of expressions including said self-tuning object in said user program;

program code for generating, in response to said detecting said plurality of expressions including said self-tuning object in said user program, a trace file indicating a sequence of said expressions including said self-tuning object in said user program;

program code for dividing said trace file into a plurality of trace file blocks;

program code for converting said trace file blocks into source code expression blocks;

program code for generating a plurality of minimal timing, compiled expression blocks, each of said plurality of minimal timing, compiled expression blocks corresponding to a respective one of said source code expression blocks, said generating said plurality of minimal timing, compiled expression blocks including application of at least one compiler optimization technique; and

program code for linking said plurality of minimal timing, compiled expression blocks into said user program.

24. A system for providing at least one self-tuning object to a user program, comprising:

means for receiving said user program;

means for simulating execution of said user program;

*Sub
Sai*

means for detecting, during said simulating of said execution of said user program, a plurality of expressions including said self-tuning object in said user program;

5 means for generating, in response to said detecting said plurality of expressions including said self-tuning object in said user program, a trace file indicating a sequence of said expressions including said self-tuning object in said user program;

10 means for dividing said trace file into a plurality of trace file blocks;

means for converting said trace file blocks into source code expression blocks;

15 means for generating a plurality of minimal timing, compiled expression blocks, each of said plurality of minimal timing, compiled expression blocks corresponding to a respective one of said source code expression blocks, said generating said plurality of minimal timing, compiled expression blocks including application of at least one compiler optimization technique; and

20 means for linking said plurality of minimal timing, compiled expression blocks into said user program.